International Journal of Innovative Computing, Information and Control
Volume **21**, Number **3**, June **2025**

ICIC International ©2025 ISSN 1349-4198
pp. **683–699**

CSCLORA: AN EFFICIENT FINE-TUNING METHOD FOR OPEN-SOURCE LARGE LANGUAGE MODELS IN CHINESE SPELLING CORRECTION

Chengzhang Li[1,2], Jun Zhou[2] and Ta Li[2,*]

[1]School of Electronic, Electrical and Communication Engineering
University of Chinese Academy of Sciences
No. 19(A), Yuquan Road, Shijingshan District, Beijing 100049, P. R. China

[2]Key Laboratory of Speech Acoustics and Content Understanding
Institute of Acoustics, Chinese Academy of Sciences
No. 21, North 4th Ring Road, Haidian District, Beijing 100190, P. R. China
{ lichengzhang; zhoujun }@hccl.ioa.ac.cn; *Corresponding author: lita@hccl.ioa.ac.cn

Received October 2024; revised February 2025

Abstract. *Chinese Spelling Correction (CSC) is used to detect and correct spelling errors in Chinese sentences automatically. It not only improves the readability and fluency of Chinese text but also avoids the interference of misspelled text on downstream tasks of Natural Language Processing (NLP). Many recent works have achieved significant results with the help of pre-trained language models represented by BERT and using data augmentation, external knowledge components, and other methods. However, for recently emerged open-source Large Language Models (LLMs), the training cost required for these advanced methods is not acceptable for individuals and small-scale laboratories. How to effectively train CSC tasks on an open-source LLM remains to be explored, which is becoming increasingly important as LLMs flourish. In this study, we offer a new universal training framework for CSC, called CSCLoRA, which allows low-cost training on publicly available LLMs. We improved the lightweight fine-tuning method Low-Rank Adaptation of LLMs (LoRA) to learn richer semantic knowledge from multiple low-dimensional representations, and only generated teacher representations required by each layer of the LLM from target texts of parallel corpus. These are automated methods that avoid data augmentation and external knowledge components relying on expert knowledge. Experiments show that our method requires comparable training costs to the original LoRA, but achieves state-of-the-art performance in open-domain and medical verticals.*

Keywords: Lightweight fine-tuning, Chinese spelling correction, Contrastive learning, LoRA, CSCLoRA, Natural language processing

1. **Introduction.** With the proliferation of the Internet and multimedia technologies, Chinese spelling errors caused by keyboard inputs, Automatic Speech Recognition (ASR), or Optical Character Recognition (OCR), have become the prevalent form of common mistakes. These errors diminish the readers' experience and increase the difficulty of accomplishing downstream tasks in NLP, including emotion recognition, entity extraction, search sorting, and automatic dialogue. Chinese Spelling Correction (CSC), which seeks to find and fix spelling mistakes in texts written in Chinese, has become prevalent in Internet text systems and is also a crucial preprocessing step for many NLP tasks [1, 2, 3].

Unlike English, Chinese is a logographic script [4], making it highly likely that there is a proximity in both pronunciation and visual form between most misspelled characters and their corresponding correct characters. According to [5], approximately 83% of misspelled

DOI: 10.24507/ijicic.21.03.683

characters are similar in pronunciation to the correct word, and 48% of misspelled characters have a visual similarity to the correct word. Humans can associate these characters with possible correct characters through association, while pre-trained language models lack the necessary training. Table 1 illustrates two examples.

TABLE 1. Two examples of misspellings caused by phonetic or visual similarities. The related correct character is indicated in bold, and the wrong character is indicated in italics. Chinese pinyin is shown in parentheses.

Phonological	Input	我起床很早，吃早餐以后坐公车到乡*虾(xia)*去了。
	Target	我起床很早，吃早餐以后坐公车到乡**下(xia)**去了。
	Translation	I woke up early and after breakfast took the bus to the countryside.
Visual	Input	我这个周*未(wei)*有空，你呢？
	Target	我这个周**末(mo)**有空，你呢？
	Translation	I'm free this weekend. How about you?

Much of the work begins by assessing the significance of characters and consolidating this information by confusing sets, related character sets, or artificially corrected datasets generated by rules [6, 7], or incorporating supplementary components to provide the model with visual and auditory data [8, 9]. These techniques can enhance the model's ability to accurately predict the likelihood of characters resembling the flawed character, consequently improving the model's performance on a larger scale. However, such methods depend on specialized expertise and may prove insufficiently comprehensive, or simply serve as a dataset optimization technique without adequate generalization.

We also find new problems when these methods are applied to large models. Larger and larger pre-trained language models perform exceptionally well across a variety of tasks but also face high adaptation costs in the case of full parameter fine-tuning. Data augmentation, external knowledge components, etc. used in advanced methods to improve performance further add to the cost. Meanwhile, large models have been adequately trained on a sufficient amount of data, so the pseudo-data generated by rule-based data augmentation, or the plug-in knowledge components with low semantic expressiveness have less effect on improving the semantic understanding of the model, and even sometimes are harmful.

For a given downstream task, it is not appropriate to fine-tune all the parameters of an LLM and generate corresponding instances of the same size as the original state-of-the-art methodology which further increases the training cost. To overcome this obstacle, a branch of research known as parameter-efficient fine-tuning has been actively developed and explored [10, 11, 12]. It is shown that optimizing only a small fraction of the parameters and keeping the pre-training model frozen can yield results that are on par with full parameter fine-tuning on many tasks.

However, existing methods for effective fine-tuning of parameters are still being explored and have mainly investigated generic frameworks. In the current paradigm, one can apply these methods to any NLP task. On the other hand, the parameter update of these fine-tuning methods still only computes the loss in the predictive layer of the model, and the existing LLMs with 40 layers (most of the open-source models with a parametric number of 13b) face a higher risk of overfitting compared to the models represented by BERT-base with only 12 layers. More research is needed on how to better train lightweight for specific downstream tasks and ensure reliable performance while reducing training costs.

These investigations have demonstrated the ability of deep language models to encode a broad range of syntactic and semantic data [13, 14, 15], and the model's upper layers depict progressively complex structures in a hierarchical manner [16, 17]. In the CSC task, which relies on the model's semantic understanding, the more correctly the multiple structural relationships of the semantics are characterized, the higher the error correction performance of the model will be. If we can give more guidance to the intermediate features of the LLM, the more likely the LLM will acquire the abstractions that, in our gut feeling, are essential for representing human language, rather than just modeling complex co-occurrence statistics on labels.

Our approach has three advantages. First, our approach generates teacher representations layer by layer, thus giving more accurate guidance to each layer of the model during training and avoiding the risk of overfitting caused by too many layers. Second, we make full use of the rich semantic knowledge embedded in the big model to generate guiding features by feeding the target text into the model instead of manually designing the dataset or components with the help of expert rules, which leads to better generalization of the error correction model. Third, our method is simple to implement and train. The intermediate layer features selected for training are rank vectors, whose dimensions are much smaller than the predictive layer representations, and the explicit memory occupancy of our method is comparable to that of the original LoRA method, whereas the adapter can be directly summed up with the corresponding matrices during inference, and the inference cost is consistent with that of the original model.

The remainder of this article is organized as follows. The next section reviews the relevant literature, and then the third section describes the algorithm design. Section 4 is devoted to presenting and discussing the experimental results. The last section summarizes the entire research work and suggests potential directions for future research.

2. Related Work.

LLM. The design of the Transformer originates from the seminal paper by Vaswani et al. [18]. Its excellent multi-head self-attention module and residual mechanism allow language models to achieve higher dimensions and deeper layers. As it has evolved over the years, the Transformer model has been commonly used in various domains and has achieved superior performance on downstream tasks such as sentiment analysis [19], translation [20], text error correction [21], and intelligent question answering [22].

With the introduction of OpenAI's model represented by ChatGPT [23], LLMs based on the Transformers structure have received great attention. These LLMs present performance in knowledge integration, fluent dialog, and logical reasoning that far exceeds the performance of previous language models represented by BERT. [24, 25] discuss their features, respectively.

In the beginning, only internal OpenAI employees or collaborators were able to participate in the development and debugging of GPT models. As research progressed, along with the development of private LLMs, academics and non-profit organizations worked on developing open-source alternatives such as LLAMA [26], Bloom [27], OPT [28], Baichuan [29], and BELLE [30]. Many of these open-source LLMs have only 13 billion or even fewer parameter counts and can run even on moderately powerful personal computers with performance close to GPT3 or even GPT3.5. This has allowed some small labs and even individuals to optimize open-source LLMs for better performance or to migrate to needed downstream tasks.

Chinese Spelling Correction. Most contemporary SOTA CSC algorithms treat the correction task as a sequence tagging task, where the correction module chooses the character that the associated token should be converted to. The disadvantage of this

strategy is that, when it comes to model inference, it considers incorrect and correct characters equally. Because of this, the model can be readily tricked by spelling characters incorrectly because it simply forecasts, based on context, which characters should be used for each slot.

To solve this problem, [31] used two BERT models as a detector and a corrector, where the detector identifies potentially incorrect locations and the corrector predicts only the correct characters in those masked locations. This approach, while avoiding the noise effect of misspelled characters by masking and making the error correction process close to the masking prediction task in the pre-training of BERT models, leads to a new problem: There is a visual or auditory similarity between the vast majority of misspelled characters and their correct counterparts, and direct masking will make correction process more difficult or even impossible.

With the emergence of language models such as BERT, the ability of neural networks to characterize deep semantics has improved dramatically, making it mainstream to use information about misspellings directly or indirectly to aid in model inference for error correction. SpellGCN feeds visual and phonological similarity information into the BERT model via an extra graph convolutional network [32]. Another notable effort is [33], which models character similarity for correction using morphological and phonetic knowledge. Similarly, [34] introduces global attention in the decoder to learn potential correlations between correctly entered and incorrect characters. Similarly, [35] designs a dynamic connectivity network to model the dependencies between neighboring characters and corrects the misspellings from contextual representations. Recently, some techniques have incorporated additional knowledge with the help of a confusion set (a collection of similar Chinese characters) or data augmentation of error-correcting datasets by rules [8, 9, 36]. These approaches allow the model to learn the correct expression that associates the corresponding error form and reduce the semantic perturbation caused by the erroneous characters.

Despite the success of these methods on models represented by BERT-base, additional components or augmented data will lead to increased training costs, making their application on LLMs expensive.

3. Approach.

3.1. Background.

Transformers. The model structure of an LLM is in the form of a decoder-only transformer. Multiple decoder layers and an embedded input layer make up the LLM model. Every layer of the decoder has a self-attentive module. The input characteristics are mapped to a collection of queries, keys, and values (referred to as q, k, v) via a linear projection layer that incorporates a weight matrix W_q, W_k, W_v. For the given q, k, v, the self-attention module is computed as

$$z = softmax\left(\frac{qk^T}{\sqrt{d_k}}\right)v$$

where z is the representation weighted by self-attention and d_k denotes the representation dimension. After that, a linear layer with a weight matrix W_o projects the output.

Low-Rank Adaptation. Neural networks usually contain a large number of fully connected layers and perform matrix multiplication to accomplish forward propagation. The parameter matrices in these fully connected layers are often full-ranked, and [12] argues that pre-trained language models tend to have a low "intrinsic dimensionality" in their parameters, which maintains excellent learning ability even when the parameter matrices to be optimized are mapped into a small subspace. With this in mind, the authors

would like to optimize only the low-rank part of the parameter matrix, and represent the optimization process of the original matrix as an optimization process of a low-rank matrix:

$$W_0 + \Delta W = W_0 + BA$$

where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$. And the rank

$$r \ll \min(d, k).$$

In this way, we can represent the forward propagation process in the following form:

$$h = W_0 x + \Delta W x = W_0 x + BAx = W_0 x + Bh_r$$

where A and B are low-dimensional matrices, and h_r is the intermediate representation after compression. By using such a transformation, the parameter scale to be optimized changes from the original parameter scale to that of a low-dimensional matrix, greatly reducing the number of parameters. The whole process is shown in Figure 1: In the LLM's training, LoRA generally adjusts only the attentional weights (W_q, W_k, W_v, W_o) and freezes all other layers, including the MLP and the normalization layer. This approach is simple and parametrically efficient. The performance of LoRA relies heavily on the expressive power of h_r, which has not been overly constrained in many previous studies, leading to low model performance.

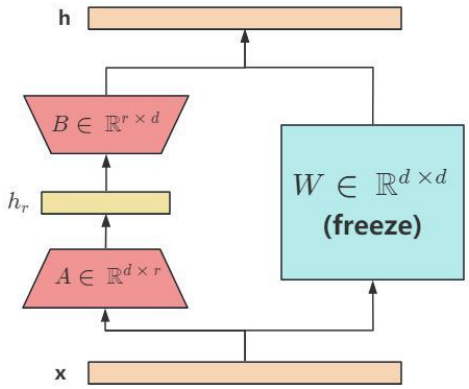

FIGURE 1. The core design of LoRA, with an adapter consisting of two low-dimensional matrices on the left and a frozen linear layer on the right

3.2. **Problem definition.** The following process can be used to characterize the CSC task. For a string of Chinese characters $X = x_1, x_2, \ldots, x_n$, detect and correct the spelling errors present in it, and output a string of Chinese characters $Y = y_1, y_2, \ldots, y_n$, where X and Y are of equal length. We generally refer to X as the input text and Y as the target text. Therefore, CSC tasks can be studied and processed as sequence labeling tasks. Usually, the percentage of spelling errors in a sentence is small, and the input text and target text are similar.

3.3. **Pilot study.** The process of fine-tuning an LLM is more challenging than that of pre-training a model, such as the BERT-base. In Table 2, we first validate the need to improve the fine-tuning methods of LLMs for CSC tasks. Even though the LLM can show comparable ability on specific tasks when it is not fine-tuned, this does not mean that the LLM is better than the BERT-base when fine-tuned on the same data. The experimental results indicate that the enhancement of the LLM appears to be relatively minor in comparison to BERT-base's capacity to learn to correct Chinese spelling errors with minimal guidance, which does not align with the substantial number of parameters

TABLE 2. Performance of BERT-base and Baichuan on CSC tasks, where Baichuan is an LLM. BERT can barely perform the error correction task without training, while the LLM has some error correction capability to begin with. However, after training BERT-base's performance backs up to the LLM.

Dataset	Model	Detection			Correction		
		Prec. (%)	Rec. (%)	F1 (%)	Prec. (%)	Rec. (%)	F1 (%)
SIGHAN15	BERT (full FT)	73.7	78.2	75.9	70.9	75.2	**73.0**
	Baichuan (w/o FT)	**88.7**	59.1	71.0	22.5	13.3	16.7
	Baichuan (FT)	59.0	**88.6**	70.8	**76.2**	67.5	71.6
	Baichuan (LoRA)	70.4	87.8	**78.2**	68.1	**76.9**	72.3

and more comprehensive pre-training of the LLM. This stems from the mismatch between the huge number of parameters of LLM and the amount of data for the downstream task, as well as the problem of vanishing gradients due to the high number of model layers. It is worth noting that the LLM performance of full-parameter fine-tuning did not significantly outperform the LLM performance using the lightweight fine-tuning approach (LoRA), which further illustrates that the traditional approach – updating all parameters of the model based on the prediction layer loss alone – leads to a greater tendency to fall into local minima in the optimization of the LLM, resulting in a model performance that does not improve with the growth of the trainable parameters.

In Table 3 we compare the training costs of BERT and LLM. It can be seen that the training time consumed by LLM after applying LoRA is comparable to that of BERT, and the occupied GPU memory is within the acceptable range for small labs or individual researchers. It is feasible and rewarding to explore adaptive methods based on LoRA that are suitable for the CSC task and thus achieve higher performance on LLM at a cost comparable to BERT training.

TABLE 3. Comparison of training costs between BERT and Baichuan for CSC task. The batch size selected for training is 16, and the adapters selected for LoRA add the objectives W_{pack} and W_o.

Model	Trainable parameters (M)	Training time (h)	Memory utilization (Gb)
BERT (full FT)	102.6	51	9.3
Baichuan (full FT)	13284.6	152	283
Baichuan (LoRA)	19.7	60	43.1

3.4. **Contrastive learning based on low-rank representations.** Existing LoRA lacks guidance for low-rank representations, and the capability of the adapter greatly depends on the semantic expressiveness of low-rank representations. When LLMs are migrated for complex downstream tasks, the optimization of the adapter will be at great risk. To solve this problem, we propose a contrastive learning method based on low-rank representations, which adaptively generates positive samples of low-rank representations from parallel data, and guides the adapter to learn the downstream complex tasks in a deep and multi-dimensional way.

The specific design is shown in Figure 2, the left side is the inference process of each module of the big model, the input characters are first converted to vector representation by embedding layer, and then step-by-step inference is carried out through N layers of

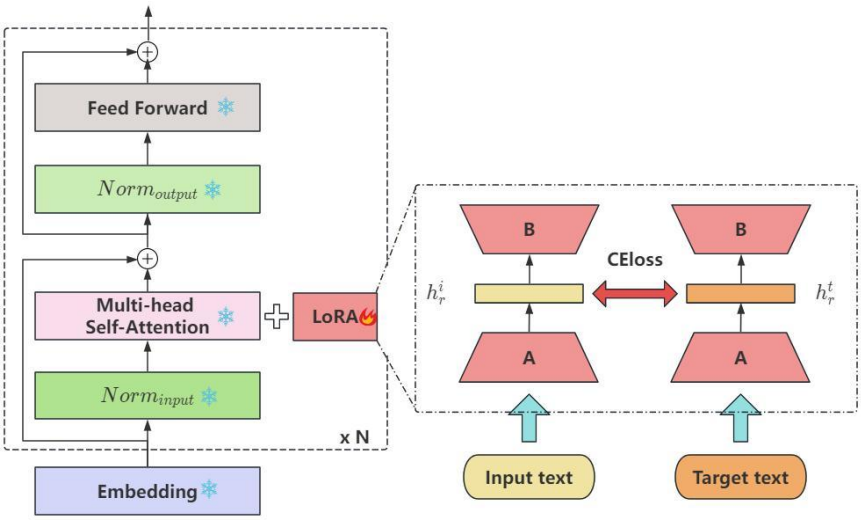

FIGURE 2. Overview of CSCLoRA design. CSCLoRA introduces contrastive learning in the fine-tuning process. In addition to the original prediction layer loss computation, CSCLoRA computes the loss on a layer-by-layer basis with the corresponding low-rank representation of the teacher, which is essential for the error correction task and takes up very little additional GPU memory.

neural network layers with the same structure. LoRA training will freeze all the above modules, and add trainable adapters on the multi-head self-attention layer. To generate h_r^t for guiding the training, we also send the target text into the model for one-time calculation instead of just being regarded as a label for calculating the prediction layer's loss. In this way, the model can generate teacher low-rank representations one-to-one for the student model at each layer. Then ultimately the cross-entropy loss between these representations is taken into account when calculating the loss for the gradient update. It should be noted that teacher representations need to be generated only during training, and our model structure is consistent with traditional LoRA, which means that like LoRA, our method does not add any additional computational cost to the large model during deployment and inference.

Attention-based Contrastive Learning. Specifically, we multidimensionally transfer language knowledge from teacher representations to student representations by adding LoRA adapters to all linear layers of the self-attention module. We add LoRA adapters to all linear layers of the self-attention module, allowing for the multidimensional transfer of linguistic knowledge from teacher representations to student representations. The linear layers referred to here are W_q, W_k, W_v, and W_o in the self-attention module. In some models, for computational efficiency, a $W_{pack} \in \mathbb{R}^{d\times 3d}$ may be used instead of $W_q \in \mathbb{R}^{d\times d}$, $W_k \in \mathbb{R}^{d\times d}$, $W_v \in \mathbb{R}^{d\times d}$, and the vectors calculated through W_{pack} are re-segmented into q, k, v.

Formally, let $H^T = \left\{H_1^T, H_2^T, \ldots, H_N^T\right\}$ be a set of teacher low-rank representations from one of the above types of linear layers (e.g., W_{pack}), and $H^S = \left\{H_1^S, H_2^S, \ldots, H_N^S\right\}$ be the student low-rank representations, where N is the number of layers with self-attention module in the LLM. Each H_i^T (resp. H_i^S) is from the i-th teacher (resp. student) attention layer. Here, we compute the distance d_i^H using cross-entropy as follows:

$$d_i^H = CE\left(H_i^S, H_i^T\right) = \sum_{j=1}^{l} CE\left(h_{ij}^S, h_{ij}^T\right) = \sum_{j=1}^{l} -softmax\left(h_{ij}^S\right) \cdot log\_softmax\left(h_{ij}^T\right) \quad (1)$$

where l is the sequence length and h_{ij}^T (resp. h_{ij}^S) denotes the j-th row of the low-rank representation matrices from H_i^T (resp. H_i^S).

Lastly, we may define the objective function for contrastive learning based on attention between H^T and H^S as

$$L_{attn} = d\left(H^S, H^T\right) = \sum_{i=1}^{N} d_i^H \tag{2}$$

where N is the LLM's total number of attention layers.

3.5. **Masked alignment of parallel data.** Although there are only substitution errors in Chinese spelling correction, this does not mean that the input text and the target text are of equal length during inference with LLMs. Unlike models such as BERT-base, which treats each Chinese character as a single token, LLMs often use Byte Pair Encoding (BPE) or Byte-level BPE (BBPE), where each token often represents a Chinese word. Therefore, in contrastive learning, we encounter the problem of input and output texts having different lengths after tokenization. As shown in Figure 3, the word '没有' in the target text is misspelled as '么有', which is tokenized into '么' and '有' by the tokenizer, resulting in an extra token in the input text compared to the output text. If we directly perform contrastive learning, the low-rank representations learned for some positions will be incorrect, which will lead to a decrease in overall model performance.

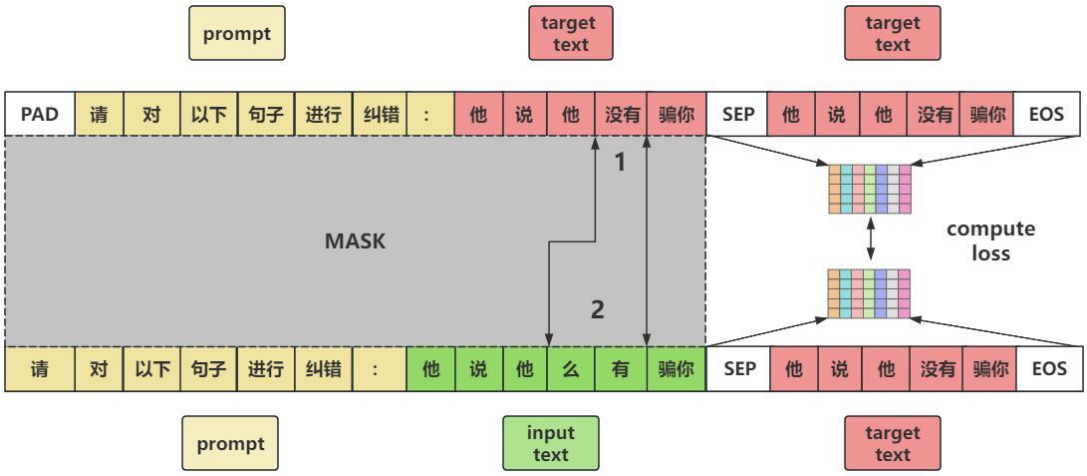

FIGURE 3. The masking alignment mechanism. This process aligns each token in the label segment sequentially for loss computation, filling the left side with <PAD> tokens to ensure proper alignment.

To solve this problem, we designed a specialized masking alignment mechanism. The input to the LLM is a string consisting of 'prompt', 'input', and 'output'. The 'prompt' is a text prompt used to inform the LLM of the current specific task, i.e., the CSC task. The 'input' is the text to be processed, which needs to be checked by the model to see if there are any spelling errors and corrected, and this part is filled in the input text and the target text, respectively, in the contrastive learning. The 'output' refers to the label, which is the text that the LLM needs to generate. This part is only used as input during training. The LLM predicts each token sequentially from front to back and calculates the loss with respect to the label.

Similar to the loss computation in the prediction layer in the regular fine-tuning task, we mask some of the low-rank representations in contrastive learning. We only compute the contrastive loss in the 'output' part, which is filled with the target text, and the

lengths of the tokenization results are aligned so that the correct representations generated from the correct utterances can be learned one-to-one. To balance the lengths of the two strings after tokenized so that they are aligned one-to-one in the 'output', we record the difference in the lengths of the input text and the target text while the split is performed, and perform appropriate padding when calculating the contrastive loss.

3.6. **Training.** In addition to learning from the middle layer features using contrastive learning, we also learn from the target text labels. As with the regular LoRA method, for the CSC task, the prediction loss can be defined as

$$L_{pred} = -softmax\left(z^T\right) \cdot log\_softmax\left(z^P\right) \tag{3}$$

where z^P denotes the probability distribution of the model output, and z^T is the one-hot label transformed from the target text label.

When training the model, we combine multiple losses with different weights to form the following overall objective:

$$L = (1-\alpha)L_{pred} + \alpha L_{attn} \tag{4}$$

where α is a hyperparameter set during training to balance the prediction loss and the middle layer's contrastive learning loss, thus improving the model performance in general.

4. **Experimental Results.** In this section, we will go over the specifics of the experiment and the key conclusions. After that, we will discuss the adapter and do more study to ensure our approach is sound.

4.1. **Experiment setup.**

Datasets. The SIGHAN dataset was derived from earlier work in [32, 37, 38], and new training data were obtained by manually annotating 10K SIGHAN samples [39, 40, 41], as well as 271K samples from [42]. We evaluate the error correction performance with the test set of the latest SIGHAN benchmark [41], as done in [31, 37, 38]. The set has 550 positive and negative samples, respectively, and contains 461 different types of errors. Negative samples are data pairs for which no spelling errors occurred, i.e., samples with the same input and output.

In addition, to verify whether we have the same state-of-the-art in the vertical domain, we also use MCSCSet [43] to check the algorithm performance. In contrast to the open-domain CSC dataset, MCSCSet gathers a substantial amount of real-world medical questions along with a large number of sentences that medical specialists have manually categorized. The dataset contains 199877 pairs of sentences, each having an average length of 10.9 characters. It is worth noting that many incorrect words in this dataset are not common in the open domain, but rather connected to medical knowledge and vocabulary.

Evaluation Metrics. The CSC task uses precision, recall, and F1 scores to comprehensively evaluate the detection and correction performance of algorithms. The following three types of evaluation algorithms dominate: character-level scores [32, 37, 42], sentence-level scores [32, 37, 44], which are evaluated using the method of [44], and scores at the phrase level assessed using SighanHan Tools, an open-source [33, 38]. Similar to the majority of earlier work, we use sentence-level scores to measure detection and correction subtasks. We believe that the error correction task is only complete if all errors in the text are corrected [44]. The specific meanings of the scores are as follows.

Sentence-level Precision: Among the source sentences that contain typos, this measures how many have every single change made correctly.

Sentence-level Recall: Among the source sentences that contain typos, this measures how many have all errors accurately corrected.

F1 Score: This is a harmonic mean of precision and recall, calculated as

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

The F1 score provides a balance between precision and recall, penalizing models that are overly precise at the expense of recall and vice versa.

Training Details. The open-source LLMs were fine-tuned with CSCLoRA to obtain a calibrated model of comparable size to the corresponding open-source model. During module training, we had a batch size of 16 and a learning rate of 1e-4. The maximum sequence length for all tasks was 256. Experiments were performed on an Nvidia A800 80GB GPU, selecting the best-performing model on the development set. Since the dimensionality of the rank is much smaller than the dimensionality of the linear layer, CSCLoRA requires very little additional GPU memory for contrastive learning. The overall GPU memory footprint of CSCLoRA is comparable to that of LoRA with the same setup. The training for the CSC task takes about 60 hours.

4.2. **Baseline.** We use the following baseline for comparison.

BERT [45]: BERT itself has semantic extraction capability. For the CSC task, predicted characters are output by adding an output layer.

Soft-Masked BERT [31]: The method uses a two-stage pipeline with two BERTs as a detector and a corrector, respectively. Through a mechanism known as soft shielding, the former and the latter are connected.

SpellGCN [32]: This method incorporates visual and phonological similarity knowledge into BERT using a customized graph.

PLOME [37]: This pretraining method combines phonological and visual features based on phonic and graphic sequences using a model that shares Bert's design.

MDCSpell [8]: The method creates a unique multitasking detector-corrector architecture in which the corrector records the phonetic and visual characteristics of every character in the source sentence using BERT.

BERT-Corrector [46]: The strategy suggests using a character replacement technique to identify more mistakes. Initially, the replacement approach is used to create the training data. Then, adversarial training and adversarial data generation are applied to lessen the impact of noise.

MedBERT-Corrector [43]: Work in the medical industry is handled by this work. The encoder PCL-MedBERT[1], a representative pre-trained medical language model, is the only distinction between the model architecture and the previously reported BERT-Corrector.

ABC-Fusion [36]: The authors propose an adapter-based approach for BERT-level obfuscated set fusion. The method dynamically extracts the relevant knowledge with the help of a lightweight adapter to semantically fuse BERT with the semantics of the obfuscated characters at the semantic encoding stage.

Baichuan-13B [29]: Open source multilingual LLM with 13 billion parameters, trained from scratch on 2.6 trillion tokens. The model is highly expressive for languages other than English, especially Chinese.

Qwen-7B [47]: Qwen is a comprehensive family of language models, including different models with varying numbers of parameters. It includes Qwen (the base pre-trained language model) and Qwen-Chat (the chat model), which is fine-tuned using human alignment techniques. The base language model consistently shows excellent performance on a wide range of downstream tasks.

[1]https://code.ihub.org.cn/projects/1775

TABLE 4. Experiments on F1 score, recall, and sentence-level precision were conducted. The outcomes indicated by a * were determined by our experiments.

Dataset	Model	Detection			Correction		
		Prec. (%)	Rec. (%)	F1 (%)	Prec. (%)	Rec. (%)	F1 (%)
SIGHAN14	BERT	65.6	68.1	66.8	63.1	65.5	64.3
	Soft-Masked *BERT**	63.7	63.2	63.5	56.7	56.2	56.4
	MDCSpell	70.2	68.8	69.5	69.0	67.7	68.3
	SpellGCN	65.1	69.5	67.2	63.1	67.2	65.3
	*PLOME**	67.4	71.5	69.4	65.3	69.3	67.2
	ABC-Fusion	68.6	**72.5**	70.6	67.1	**70.7**	68.9
	Qwen (CSCLoRA)	74.3	70.2	72.2	71.6	68.8	69.7
	Baichuan (CSCLoRA)	**75.2**	71.3	**73.2**	**75.0**	70.3	**72.6**
SIGHAN15	BERT	73.7	78.2	75.9	70.9	75.2	73.0
	Soft-Masked BERT	73.7	73.2	73.5	66.7	66.2	66.4
	MDCSpell	80.8	80.6	80.7	78.4	78.2	78.3
	SpellGCN	74.8	80.7	77.7	72.1	77.7	75.9
	PLOME	77.4	81.5	79.4	75.3	79.3	77.2
	ABC-Fusion	79.5	82.3	80.9	77.4	80.1	78.7
	Qwen (CSCLoRA)	83.9	79.0	81.4	80.8	78.2	79.5
	Baichuan (CSCLoRA)	**87.8**	**84.3**	**85.1**	**83.9**	**81.1**	**82.5**
MCSCSet	*BERT**	85.2	84.2	84.7	78.8	78.2	78.5
	BERT-Corrector	87.1	86.1	86.6	80.9	80.1	80.5
	Soft-Masked BERT	87.0	86.3	86.7	81.2	80.5	80.9
	MedBERT-Corrector	87.0	86.3	86.6	81.0	80.2	80.6
	*PLOME**	88.8	88.3	88.6	83.1	82.6	82.8
	Qwen (CSCLoRA)	93.3	91.1	92.2	84.7	82.5	83.6
	Baichuan (CSCLoRA)	**95.3**	**93.9**	**94.6**	**85.7**	**84.5**	**85.1**

4.3. **Main results.** We used each of the three training sets described above and examined the effects on the corresponding test sets. Our approach is a generic CSC method that can be used for open-source LLMs, and for this purpose, two representative open-source LLMs, Qwen and Baichuan, are selected to test the performance of the algorithm. Table 4 presents our experimental findings together with a comparison to earlier findings on the same dataset. Based on the findings, we can note that

1) Both on the detection and correction tasks, our system performs much better than the SOTA baseline method. The introduction of contrastive learning in LoRA allows our method to better learn the key semantics needed for the CSC task and avoid overfitting during training. For example, Baichuan's experiments on the dataset SIGHAN showed that the F1 scores improved by 2.6% and 4.2% in the detection task, and 3.7% and 3.8% in the correction task, respectively. These outcomes show the effectiveness of CSCLoRA.

2) Previous studies have relied on a priori knowledge provided by experts and have additionally input a variety of supplementary information during training and inference to improve correction performance. For example, SpellGCN and ABC-Fusion construct confusing sets based on character similarity, while MDCSpell and PLOME utilize multimodal information and expert rule-based data augmentation. Unlike these approaches, CSCLoRA does not require any a priori knowledge; instead, competitive performance is obtained by introducing contrastive learning, mining the original semantic knowledge in the pre-trained LLM, and migrating it to the downstream CSC task.

3) The experimental results on MCSCSet also indirectly confirm the idea in 2). Even if the open domain is replaced with the medical domain as the target domain, our approach still operates fairly well. This is due to the large amount of knowledge acquired by LLM during the pre-training process, and also since our adaptive method is free from the dependence on a priori knowledge. Other methods, however, are misled by low-frequency errors or lack of relevant domain knowledge, resulting in performance degradation. MedBERT-Corrector uses data containing medical domain knowledge for pre-training, but its correction task's F1 performance is 4.5% lower than that of our method. This indicates that the error correction task benefits less from vertical domain semantic knowledge and that, to help the model avoid being misled by the semantics of erroneous characters during the prediction process, contrastive learning should be used to introduce the full semantics of correct sentences during error correction training.

4.4. **Ablation study and discussions.**

Ablation Study. To validate the effects of contrastive learning and masked alignment mechanisms on model performance, we conducted an ablation study. Our CSCLoRA requires the addition of adapters for the positions W_{pack} and W_o. In this ablation study, we will cancel the contrastive learning loss at the specified location, which is equivalent to the adapter at that location being trained according to the traditional LoRA. Table 5 shows the inference performance of our model on two datasets with different settings. We found several things.

1) Even if only a portion of the comparison learning module is retained, it allows for a large improvement in model performance relative to the base LoRA method. This implies that introducing semantic representations of correct utterances into training via contrastive learning is crucial for the CSC task.

2) Contrastive learning for both W_{pack} and W_o contributes roughly the same inference performance; after all, both are part of the multi-head attention module. However, we also notice that removing the contrastive learning of W_{pack} always loses more performance, which stems from the fact that W_{pack} is a splicing matrix of Q, K, and V, which has a

TABLE 5. Ablation study of contrastive learning and masked alignment mechanism introduced in CSCLoRA based on Baichuan-13B. -W_{pack} or -W_o means that the linear layer at the corresponding position does not compute the contrastive learning loss. -Alignment means that instead of masked alignment each token of the input text is directly compared one-to-one for contrastive learning.

Dataset	Model	Detection			Correction		
		Prec. (%)	Rec. (%)	F1 (%)	Prec. (%)	Rec. (%)	F1 (%)
SIGHAN14	LoRA	65.6	68.1	66.8	63.1	65.5	64.3
	CSCLoRA	75.2	71.3	73.2	75.0	70.3	72.6
	-W_{pack}	71.5	68.8	70.1	71.0	67.8	69.4
	-W_o	72.8	69.5	71.1	72.9	67.6	70.7
	-Alignment	69.4	65.5	67.4	65.8	62.2	64.0
SIGHAN15	LoRA	70.4	87.8	78.2	68.1	76.9	72.3
	CSCLoRA	87.8	84.3	85.1	83.9	81.1	82.5
	-W_{pack}	83.2	80.8	82.0	78.2	76.5	77.3
	-W_o	84.6	81.7	82.6	80.6	76.9	78.6
	-Alignment	74.7	78.7	76.7	68.3	72.6	70.4

greater impact on the distribution in the attention matrix, and the attention to the key subwords in turn has a direct impact on the effect of error detection and correction.

3) With the removal of the masked alignment mechanism, contrastive learning loses its effect almost completely, and the performance of CSCLoRA is not much different from LoRA. Although some tokens are not aligned due to the different expressions of correct and incorrect texts, some tokens are aligned, and even some of the parallel corpus are aligned one to one after tokenized, such as the negative samples in the dataset (with consistent inputs and outputs). In this case, the negative benefit due to partially misaligned tokens offsets the positive benefit gained from correct contrastive learning training with aligned tokens.

Impact of the rank r. We study the effect of rank r on model performance and the results are shown in Table 6. It can be seen that the performance of the model reaches its maximum value after r is greater than 4. Surprisingly even with r = 1 the model still has quite a high performance. This is in line with LoRA's assumption that the adapter matrix ΔW may have a very small "intrinsic rank" and that a small rank chosen by the adaptive module is sufficient.

TABLE 6. The F1 (%) of the error correction model trained with different ranks r on the test set

Dataset	Task	r				
		1	2	4	8	16
SIGHAN14	Detection	72.1	72.3	73.0	73.2	73.1
	Correction	71.8	72.4	72.6	72.6	72.6
SIGHAN15	Detection	84.5	84.7	85.2	85.1	84.9
	Correction	81.7	82.0	82.4	82.5	82.4

Case Studies. Table 7 presents three samples chosen from the test set to illustrate the efficacy of our targeted enhancements to LoRA. These examples demonstrate how contrastive learning helps the model learn to mine the correlation between the words that have spelling errors and the correct words, thereby removing noise from the model's semantic understanding of the input utterances.

1) While LoRA detects two errors in the SIGHAN14 example, it only fixes one error that contains two characters. This is because the more characters there are in a row of errors, the harder it is for the model to understand the semantics of the input text and thus give up on correcting that part of the text. Contrastive learning facilitates the model's understanding of the character changes brought about by common errors, enabling the model to comprehend the general semantics of the input text that contains errors.

2) In SIGHAN15's example, LoRA does not correct the misspelled character, even though it is pronounced the same as the correct character. Associations between candidate words are easy for humans but not often so for models, which are confronted with a somewhat cleaned standard corpus during pre-training. Our approach complements this training in associative relations to some extent.

3) The examples in MCSCSet involve medical expertise. The model obtained from LoRA training checked for errors but did not modify them correctly. However, when we masked out the misspelled words and handed them directly to the model for prediction, the model output the correct characters. This demonstrates that while misspelled characters provide information on phonetic proximity to help inference, their own semantic information also interferes with the model's inference. Our approach avoids the above problem by forcing the model to approximate the semantic representations of the incorrect and correct characters during training.

TABLE 7. In three instances taken from various datasets, CSCLoRA outperforms LoRA by a significant margin.

SIGHAN14	Input	可是很多外国人很喜欢*赛太杨*在海边，在印尼有很多外国人来*路性*。
	Target	可是很多外国人很喜欢**晒太阳**在海边，在印尼有很多外国人来**旅行**。
	Translation	However, many foreigners love to sunbathe on the beach, and there are many foreigners traveling in Indonesia.
	LoRA	可是很多外国人很喜欢*赛太杨*在海边，在印尼有很多外国人来**旅行**。
	CSCLoRA	可是很多外国人很喜欢**晒太阳**在海边，在印尼有很多外国人来**旅行**。
SIGHAN15	Input	他很少*座(zuo)*捷运去玩。
	Target	他很少**坐(zuo)**捷运去玩。
	Translation	He rarely takes the MRT for fun.
	LoRA	他很少*座(zuo)*捷运去玩。
	CSCLoRA	他很少**坐(zuo)**捷运去玩。
MCSCSet	Input	宫*径(jing)*癌术后排尿锻练的方法
	Target	宫**颈(jing)**癌术后排尿锻练的方法
	Translation	Methods of urine voiding exercise after cervical cancer surgery
	LoRA	宫*紧(jin)*癌术后排尿锻练的方法
	CSCLoRA	宫**颈(jing)**癌术后排尿锻练的方法

5. **Conclusion.** In this paper, we design a novel parameter-efficient fine-tuning method for CSC tasks, which enables LLMs to achieve competitive performance at an acceptable training cost by designing contrast learning and masking alignment mechanisms. Our training method does not need to rely on external knowledge and components, can adaptively extract key semantics for open-source LLMs trained on CSC tasks, and avoids the overfitting phenomenon caused by too many parameters through layer-by-layer learning. Through extensive experiments, we demonstrate the effectiveness and practicality of our approach. We plan to explore multi-dimensional learning methods for more modules in LLM in the next step, and how to better generate teacher representations used as positive samples to further improve the performance of the model. In addition, we will investigate whether our core ideas are equally applicable to other NLP downstream tasks.

Acknowledgment. The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] K.-Y. Chen, H.-S. Lee, C.-H. Lee, H.-M. Wang and H.-H. Chen, A study of language modeling for Chinese spelling check, *Proc. of the 7th SIGHAN Workshop on Chinese Language Processing*, pp.79-83, 2013.

[2] J. Yu and Z. Li, Chinese spelling error detection and correction based on language model, pronunciation, and shape, *Proc. of the 3rd CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp.220-223, 2014.

[3] C. Aswin and W. A. Wella, Bidirectional encoder representations from transformers for cyberbullying text detection in indonesian social media, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1599-1615, 2021.

[4] G. Sampson, *Writing Systems*, H Utchinson, London, UK, 1985.

[5] C.-L. Liu, M.-H. Lai, Y.-H. Chuang and C.-Y. Lee, Visually and phonologically similar characters in incorrect simplified Chinese words, *Coling 2010: Posters*, pp.739-747, 2010.
[6] D. Wang, Y. Song, J. Li, J. Han and H. Zhang, A hybrid approach to automatic corpus generation for Chinese spelling check, *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp.2517-2527, 2018.
[7] D. Wang, Y. Tay and L. Zhong, Confusionset-guided pointer networks for Chinese spelling check, *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp.5780-5785, 2019.
[8] C. Zhu, Z. Ying, B. Zhang and F. Mao, MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction, *Findings of the Association for Computational Linguistics: ACL 2022*, pp.1244-1253, 2022.
[9] S. Liu, S. Song, T. Yue, T. Yang, H. Cai, T. Yu and S. Sun, CRASpell: A contextual typo robust approach to improve Chinese spelling correction, *Findings of the Association for Computational Linguistics: ACL 2022*, pp.3008-3018, 2022.
[10] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal and C. Raffel, Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, *Advances in Neural Information Processing Systems*, vol.35, pp.1950-1965, 2022.
[11] X. L. Li and P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, *arXiv Preprint*, arXiv: 2101.00190, 2021.
[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang and W. Chen, LoRA: Low-rank adaptation of large language models, *arXiv Preprint*, arXiv: 2106.09685, 2021.
[13] X. Shi, I. Padhi and K. Knight, Does string-based neural MT learn source syntax?, *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp.1526-1534, 2016.
[14] Y. Belinkov, *On Internal Language Representations in Deep Learning: An Analysis of Machine Translation and Speech Recognition*, Ph.D. Thesis, Massachusetts Institute of Technology, 2018.
[15] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. Van Durme, S. R. Bowman, D. Das et al., What do you learn from context? Probing for sentence structure in contextualized word representations, *arXiv Preprint*, arXiv: 1905.06316, 2019.
[16] M. E. Peters, M. Neumann, L. Zettlemoyer and W.-T. Yih, Dissecting contextual word embeddings: Architecture and representation, *arXiv Preprint*, arXiv: 1808.08949, 2018.
[17] T. Blevins, O. Levy and L. Zettlemoyer, Deep RNNs encode soft hierarchical syntax, *arXiv Preprint*, arXiv: 1805.04218, 2018.
[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, *Advances in Neural Information Processing Systems*, vol.30, 2017.
[19] W. Zhang, Y. Deng, B. Liu, S. J. Pan and L. Bing, Sentiment analysis in the era of large language models: A reality check, *arXiv Preprint*, arXiv: 2305.15005, 2023.
[20] A. Hendy, M. Abdelrehim, A. Sharaf, V. Raunak, M. Gabr, H. Matsushita, Y. J. Kim, M. Afify and H. H. Awadalla, How good are GPT models at machine translation? A comprehensive evaluation, *arXiv Preprint*, arXiv: 2302.09210, 2023.
[21] Y. Fan, F. Jiang, P. Li and H. Li, GrammarGPT: Exploring open-source LLMs for native Chinese grammatical error correction with supervised fine-tuning, *CCF International Conference on Natural Language Processing and Chinese Computing*, pp.69-80, 2023.
[22] X. L. Dong, S. Moon, Y. E. Xu, K. Malik and Z. Yu, Towards next-generation intelligent assistants leveraging LLM techniques, *Proc. of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp.5792-5793, 2023.
[23] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han and Y. Tang, A brief overview of ChatGPT: The history, status quo and potential future development, *IEEE/CAA Journal of Automatica Sinica*, vol.10, no.5, pp.1122-1136, 2023.
[24] X. Chen, J. Ye, C. Zu, N. Xu, R. Zheng, M. Peng, J. Zhou, T. Gui, Q. Zhang and X. Huang, How robust is GPT-3.5 to predecessors? A comprehensive study on language understanding tasks, *arXiv Preprint*, arXiv: 2303.00293, 2023.
[25] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg et al., Sparks of artificial general intelligence: Early experiments with GPT-4, *arXiv Preprint*, arXiv: 2303.12712, 2023.
[26] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar et al., LLaMA: Open and efficient foundation language models, *arXiv Preprint*, arXiv: 2302.13971, 2023.

[27] BigScience Workshop, T. Le Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon et al., BLOOM: A 176B-parameter open-access multilingual language model, *arXiv Preprint*, arXiv: 2211.05100, 2022.

[28] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin et al., OPT: Open pre-trained transformer language models, *arXiv Preprint*, arXiv: 2205.01068, 2022.

[29] A. Yang, B. Xiao, B. Wang, B. Zhang, C. Bian, C. Yin, C. Lv, D. Pan, D. Wang, D. Yan et al., Baichuan 2: Open large-scale language models, *arXiv Preprint*, arXiv: 2309.10305, 2023.

[30] Y. Ji, Y. Gong, Y. Deng, Y. Peng, Q. Niu, B. Ma and X. Li, Towards better instruction following language models for Chinese: Investigating the impact of training data and evaluation, *arXiv Preprint*, arXiv: 2304.07854, 2023.

[31] S. Zhang, H. Huang, J. Liu and H. Li, Spelling error correction with Soft-Masked BERT, *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pp.882-890, 2020.

[32] X. Cheng, W. Xu, K. Chen, S. Jiang, F. Wang, T. Wang, W. Chu and Y. Qi, SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check, *arXiv Preprint*, arXiv: 2004.14166, 2020.

[33] L. Huang, J. Li, W. Jiang, Z. Zhang, M. Chen, S. Wang and J. Xiao, PHMOSpell: Phonological and morphological knowledge guided Chinese spelling check, *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp.5958-5967, 2021.

[34] Z. Guo, Y. Ni, K. Wang, W. Zhu and G. Xie, Global attention decoder for Chinese spelling error correction, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp.1419-1428, 2021.

[35] B. Wang, W. Che, D. Wu, S. Wang, G. Hu and T. Liu, Dynamic connected networks for Chinese spelling check, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp.2437-2446, 2021.

[36] J. Xie, K. Dang, J. Liu and E. Liang, ABC-Fusion: Adapter-based BERT-level confusion set fusion approach for Chinese spelling correction, *Computer Speech & Language*, vol.83, 101540, 2023.

[37] S. Liu, T. Yang, T. Yue, F. Zhang and D. Wang, PLOME: Pre-training with misspelled knowledge for Chinese spelling correction, *Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp.2991-3000, 2021.

[38] P. Li and S. Shi, Tail-to-tail non-autoregressive sequence prediction for Chinese grammatical error correction, *arXiv Preprint*, arXiv: 2106.01609, 2021.

[39] S.-H. Wu, C.-L. Liu and L.-H. Lee, Chinese spelling check evaluation at SIGHAN bake-off 2013, *Proc. of the 7th SIGHAN Workshop on Chinese Language Processing*, pp.35-42, 2013.

[40] L.-C. Yu, L.-H. Lee, Y.-H. Tseng and H.-H. Chen, Overview of SIGHAN 2014 bake-off for Chinese spelling check, *Proc. of the 3rd CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp.126-132, 2014.

[41] Y.-H. Tseng, L.-H. Lee, L.-P. Chang and H.-H. Chen, Introduction to SIGHAN 2015 bake-off for Chinese spelling check, *Proc. of the 8th SIGHAN Workshop on Chinese Language Processing*, pp.32-37, 2015.

[42] D. Wang, Y. Song, J. Li, J. Han and H. Zhang, A hybrid approach to automatic corpus generation for Chinese spelling check, *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp.2517-2527, 2018.

[43] W. Jiang, Z. Ye, Z. Ou, R. Zhao, J. Zheng, Y. Liu, B. Liu, S. Li, Y. Yang and Y. Zheng, MCSCSet: A specialist-annotated dataset for medical-domain Chinese spelling correction, *Proc. of the 31st ACM International Conference on Information & Knowledge Management*, pp.4084-4088, 2022.

[44] Y. Hong, X. Yu, N. He, N. Liu and J. Liu, FASPell: A fast, adaptable, simple, powerful Chinese spell checker based on DAE-decoder paradigm, *Proc. of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019)*, pp.160-169, 2019.

[45] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *arXiv Preprint*, arXiv: 1810.04805, 2018.

[46] C. Li, C. Zhang, X. Zheng and X. Huang, Exploration and exploitation: Two ways to improve Chinese spelling correction models, *arXiv Preprint*, arXiv: 2105.14813, 2021.

[47] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang et al., Qwen technical report, *arXiv Preprint*, arXiv: 2309.16609, 2023.

Author Biography

Chengzhang Li received a Bachelor of Engineering degree from Information Science and Technology, Tsinghua University, Beijing, China, in 2019, and is currently pursuing a Ph.D. Signal and Information Processing from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China. His research interests focus on natural language processing and large models such as RAG (Retrieval-Augmented Generation). He is dedicated to exploring the frontiers of language understanding and the practical applications of large-scale AI models.

Jun Zhou obtained his Ph.D. degree in Computer Software and Theory from the Institute of Information Engineering, Chinese Academy of Sciences, China in 2017. He is now working as a Researcher at the Institute of Acoustics, Chinese Academy of Sciences. His main research interest is natural language processing such as semantic retrieval and large language model.

Ta Li received the B.S. degree in Electrical Engineering from Nanjing University, Nanjing, China, in 2003 and the Ph.D. degree in Signal and Information Processing from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, in 2010. He is currently a Professor with the Key Laboratory of Speech Acoustics and Content Understanding, Chinese Academy of Sciences. His research interests include spontaneous speech recognition, keyword spotting, and embedded speech recognition systems.